Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 1, 2017

X. Liu
Kuatro Technologies
Y. Qu
A. Lindem
Cisco Systems
C. Hopps
Deutsche Telekom
L. Berger
LabN Consulting, L.L.C.
October 28, 2016

Routing Area Common YANG Data Types
draft-rtgyangdt-rtgwg-routing-types-00

Abstract

This document defines a collection of common data types using YANG data modeling language. These derived common types are designed to be imported by other modules defined in the routing area.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

YANG [RFC6020] [RFC7950] is a data modeling language used to model configuration data, state data, Remote Procedure Calls, and notifications for network management protocols. The YANG language supports a small set of built-in data types and provides mechanisms to derive other types from the built-in types.

This document introduces a collection of common data types derived from the built-in YANG data types. The derived types are designed to be the common types applicable for modeling in the routing area.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

1.2. Terminology

The terminology for describing YANG data models is found in [RFC7950].

2. Overview

This document defines the following data types:

router-id
Router Identifiers are commonly used to identify a nodes in routing and other control plane protocols. An example usage of router-id can be found in [I-D.ietf-ospf-yang].

address-family
This type defines values for use in address family identifiers. The values are based on the IANA Address Family Numbers Registry [1]. An example usage can be found in [I-D.ietf-idr-bgp-model].

route-target
Route Targets (RTs) are commonly used to control the distribution of virtual routing and forwarding (VRF) information, see [RFC4364], in support of virtual private networks (VPNs). An example usage can be found in [I-D.ietf-idr-bgp-model] and

route-distinguisher
Route Distinguishers (RDs) are commonly used to identify separate routes in support of virtual private networks (VPNs). For example, in [RFC4364], RDs are commonly used to identify independent VPNs and VRFs, and more generally, to identify multiple routes to the same prefix. An example usage can be found in [I-D.ietf-idr-bgp-model].

ieee-bandwidth
Bandwidth in IEEE 754 floating point 32-bit binary format [IEEE754]. Commonly used in Traffic Engineering control plane protocols. An example of where this type may/will be used is [I-D.ietf-ospf-yang].

link-access-type
This type identifies the IGP link type. An example of where this type may/will be used is [I-D.ietf-ospf-yang].

multicast-source-ipv4-addr-type
IPv4 source address type for use in multicast control protocols. This type also allows the indication of wildcard sources, i.e., "*". An example of where this type may/will be used is [I-D.ietf-pim-yang].

multicast-source-ipv6-addr-type
IPv6 source address type for use in multicast control protocols. This type also allows the indication of wildcard sources, i.e.,

"*". An example of where this type may/will be used is [I-D.ietf-pim-yang].

timer-multiplier
This type is used in conjunction with a timer-value type. It is generally used to indicate define the number of timer-value intervals that may expire before a specific event must occur. Examples of this include the arrival of any BFD packets, see [RFC5880] Section 6.8.4, or hello_interval in [RFC3209]. Example of where this type may/will be used is [I-D.ietf-idr-bgp-model] and [I-D.ietf-teas-yang-rsvp].

timer-value-seconds16
This type covers timers which can be set in seconds, not set, or set to infinity. This type supports a range of values that can be represented in a uint16 (2 octets). An example of where this type may/will be used is [I-D.ietf-ospf-yang].

timer-value-seconds32
This type covers timers which can be set in seconds, not set, or set to infinity. This type supports a range of values that can be represented in a uint32 (4 octets). An example of where this type may/will be used is [I-D.ietf-teas-yang-rsvp].

timer-value-milliseconds
This type covers timers which can be set in milliseconds, not set, or set to infinity. This type supports a range of values that can be represented in a uint32 (4 octets). Examples of where this type may/will be used include [I-D.ietf-teas-yang-rsvp] and [I-D.ietf-bfd-yang].

3. YANG Module

```
<CODE BEGINS> file "ietf-routing-types@2016-10-28.yang"
module ietf-routing-types {

  namespace "urn:ietf:params:xml:ns:yang:ietf-routing-types";
  prefix "rt-types";

  import ietf-yang-types {
    prefix "yang";
  }

  import ietf-inet-types {
    prefix "inet";
  }

  organization "IETF Routing Area Working Group (rtgwg)";
```

```
     contact
         "Routing Area Working Group - <rtgwg@ietf.org>";

     description
       "This module contains a collection of YANG data types
        considered generally useful for routing protocols.";

     revision 2016-10-28 {
       description
         "Initial revision.";
       reference
         "RFC TBD: Routing YANG Data Types";
     }

     /*** collection of types related to routing ***/
     typedef router-id {
       type yang:dotted-quad;
       description
         "A 32-bit number in the dotted quad format assigned to each
          router. This number uniquely identifies the router within an
          Autonomous System.";
     }

     // address-family
     identity address-family {
       description
         "Base identity from which identities describing address
          families are derived.";
     }

     identity ipv4 {
       base address-family;
       description
         "This identity represents IPv4 address family.";
     }

     identity ipv6 {
       base address-family;
       description
         "This identity represents IPv6 address family.";
     }

     //The rest of the values deinfed in the IANA registry

     identity nsap {
       base address-family;
       description
         "Address family from IANA registry.";
```

```
     }
     identity hdlc {
       base address-family;
       description
         "(8-bit multidrop)
           Address family from IANA registry.";
     }
     identity bbn1822 {
       base address-family;
       description
         "AHIP (BBN report #1822)
          Address family from IANA registry.";
     }
     identity ieee802 {
       base address-family;
       description
         "(includes all 802 media plus Ethernet canonical format)
          Address family from IANA registry.";
     }
     identity e163 {
       base address-family;
       description
         "Address family from IANA registry.";
     }
     identity e164 {
       base address-family;
       description
         "SMDS, Frame Relay, ATM
          Address family from IANA registry.";
     }
     identity f69 {
       base address-family;
       description
         "(Telex)
          Address family from IANA registry.";
     }
     identity x121 {
       base address-family;
       description
         "(X.25, Frame Relay)
          Address family from IANA registry.";
     }
     identity ipx {
       base address-family;
       description
         "Address family from IANA registry.";
     }
     identity appletalk {
```

```
        base address-family;
        description
          "Address family from IANA registry.";
      }
      identity decnet-iv {
        base address-family;
        description
          "Decnet IV
           Address family from IANA registry.";
      }
      identity vines {
        base address-family;
        description
          "Banyan Vines
           Address family from IANA registry.";
      }
      identity e164-nsap {
        base address-family;
        description
          "E.164 with NSAP format subaddress
           Address family from IANA registry.";
      }
      identity dns {
        base address-family;
        description
          "Domain Name System
           Address family from IANA registry.";
      }
      identity dn {
        base address-family;
        description
          "Distinguished Name
           Address family from IANA registry.";
      }
      identity as-num {
        base address-family;
        description
          "AS Number
           Address family from IANA registry.";
      }
      identity xtp-v4 {
        base address-family;
        description
          "XTP over IPv4
           Address family from IANA registry.";
      }
      identity xtp-v6 {
        base address-family;
```

```
       description
         "XTP over IPv6
          Address family from IANA registry.";
     }
     identity xtp {
       base address-family;
       description
         "XTP native mode XTP
          Address family from IANA registry.";
     }
     identity fc-port {
       base address-family;
       description
         "Fibre Channel World-Wide Port Name
          Address family from IANA registry.";
     }
     identity fc-node {
       base address-family;
       description
         "Fibre Channel World-Wide Node Name
          Address family from IANA registry.";
     }
     identity gwid {
       base address-family;
       description
         "Address family from IANA registry.";
     }
     identity l2vpn {
       base address-family;
       description
         "Address family from IANA registry.";
     }
     identity mpls-tp-section-eid {
       base address-family;
       description
         "MPLS-TP Section Endpoint Identifier
          Address family from IANA registry.";
     }
     identity mpls-tp-lsp-eid {
       base address-family;
       description
         "MPLS-TP LSP Endpoint Identifier
          Address family from IANA registry.";
     }
     identity mpls-tp-pwe-eid {
       base address-family;
       description
         "MPLS-TP Pseudowire Endpoint Identifier
```

```
          Address family from IANA registry.";
     }
     identity mt-v4 {
       base address-family;
       description
         "Multi-Topology IPv4.
          Address family from IANA registry.";
     }
     identity mt-v6 {
       base address-family;
       description
         "Multi-Topology IPv6.
          Address family from IANA registry.";
     }

     /*** collection of types related to VPN ***/
     typedef route-target {
       type string {
         pattern
           '(0:(6553[0-5]|655[0-2]\d|65[0-4]\d{2}|6[0-4]\d{3}|'
         + '[0-5]?\d{0,3}\d):(429496729[0-5]|42949672[0-8]\d|'
         + '4294967[01]\d{2}|429496[0-6]\d{3}|42949[0-5]\d{4}|'
         + '4294[0-8]\d{5}|429[0-3]\d{6}|42[0-8]\d{7}|4[01]\d{8}|'
         + '[0-3]?\d{0,8}\d))|'
         + '(1:(((\d|[1-9]\d|1\d{2}|2[0-4]\d|25[0-5])\.){3}(\d|[1-9]\d|'
         + '1\d{2}|2[0-4]\d|25[0-5])):(6553[0-5]|655[0-2]\d|'
         + '65[0-4]\d{2}|6[0-4]\d{3}|[0-5]?\d{0,3}\d))|'
         + '(2:(429496729[0-5]|42949672[0-8]\d|4294967[01]\d{2}|'
         + '429496[0-6]\d{3}|42949[0-5]\d{4}|4294[0-8]\d{5}|'
         + '429[0-3]\d{6}|42[0-8]\d{7}|4[01]\d{8}|[0-3]?\d{0,8}\d):'
         + '(6553[0-5]|655[0-2]\d|65[0-4]\d{2}|6[0-4]\d{3}|'
         + '[0-5]?\d{0,3}\d))';
       }
       description
         "Route target has a similar format to route distinguisher.
          A route target consists of three fields:
          a 2-byte type field, an administrator field,
          and an assigned number field.
          According to the data formats for type 0, 1, and 2 defined in
          RFC4360, the encoding pattern is defined as:

          0:2-byte-asn:4-byte-number
          1:4-byte-ipv4addr:2-byte-number
          2:4-byte-asn:2-byte-number.

          Some valid examples are: 0:100:100, 1:1.1.1.1:100, and
          2:1234567890:203.";
       reference
```

```
         "RFC4360: BGP Extended Communities Attribute.";
     }

     typedef route-distinguisher {
       type string {
         pattern
           '(0:(6553[0-5]|655[0-2]\d|65[0-4]\d{2}|6[0-4]\d{3}|'
         + '[0-5]?\d{0,3}\d):(429496729[0-5]|42949672[0-8]\d|'
         + '4294967[01]\d{2}|429496[0-6]\d{3}|42949[0-5]\d{4}|'
         + '4294[0-8]\d{5}|429[0-3]\d{6}|42[0-8]\d{7}|4[01]\d{8}|'
         + '[0-3]?\d{0,8}\d))|'
         + '(1:(((\d|[1-9]\d|1\d{2}|2[0-4]\d|25[0-5])\.){3}(\d|[1-9]\d|'
         + '1\d{2}|2[0-4]\d|25[0-5])):(6553[0-5]|655[0-2]\d|'
         + '65[0-4]\d{2}|6[0-4]\d{3}|[0-5]?\d{0,3}\d))|'
         + '(2:(429496729[0-5]|42949672[0-8]\d|4294967[01]\d{2}|'
         + '429496[0-6]\d{3}|42949[0-5]\d{4}|4294[0-8]\d{5}|'
         + '429[0-3]\d{6}|42[0-8]\d{7}|4[01]\d{8}|[0-3]?\d{0,8}\d):'
         + '(6553[0-5]|655[0-2]\d|65[0-4]\d{2}|6[0-4]\d{3}|'
         + '[0-5]?\d{0,3}\d))|'
         + '(([3-9a-fA-F]|[1-9a-fA-F][\da-fA-F]{1,3}):'
         + '[\da-fA-F]{1,12})';
       }
       description
         "Route distinguisher has a similar format to route target.
          An route distinguisher consists of three fields:
          a 2-byte type field, an administrator field,
          and an assigned number field.
          According to the data formats for type 0, 1, and 2 defined in
          RFC4364, the encoding pattern is defined as:

          0:2-byte-asn:4-byte-number
          1:4-byte-ipv4addr:2-byte-number
          2:4-byte-asn:2-byte-number.
          2-byte-other-hex-number:6-byte-hex-number

          Some valid examples are: 0:100:100, 1:1.1.1.1:100, and
          2:1234567890:203.";
       reference
         "RFC4364: BGP/MPLS IP Virtual Private Networks (VPNs).";
     }

     /*** collection of types common to protocols ***/
     typedef ieee-bandwidth {
       type string {
         pattern
           '0[xX](0((\.0?)?[pP](\+)?0?|(\.0?))|'
         + '1(\.([\da-fA-F]{0,5}[02468aAcCeE]?)?)?[pP](\+)?(12[0-7]|'
         + '1[01]\d|0?\d?\d)?)|0[xX][\da-fA-F]{1,8}';
```

```
      }
      description
        "Bandwidth in IEEE 754 floating point 32-bit binary format:
         (-1)**(S) * 2**(Exponent-127) * (1 + Fraction),
         where Exponent uses 8 bits, and Fraction uses 23 bits.
         The units are bytes per second.
         The encoding format is the external hexadecimal-significand
         character sequences specified in IEEE 754 and C99,
         restricted to be normalized, non-negative, and non-fraction:
         0x1.hhhhhhp{+}d or 0X1.HHHHHHP{+}D
         where 'h' and 'H' are hexadecimal digits, 'd' and 'D' are
         integers in the range of [0..127].
         When six hexadecimal digits are used for 'hhhhhh' or 'HHHHHH',
         the least significant digit must be an even number.
         'x' and 'X' indicate hexadecimal; 'p' and 'P' indicate power
         of two.
         Some examples are: 0x0p0, 0x1p10, and 0x1.abcde2p+20";
      reference
        "IEEE Std 754-2008: IEEE Standard for Floating-Point
         Arithmetic.";
    }

    typedef link-access-type {
      type enumeration {
        enum "broadcast" {
          description
          "Specify broadcast multi-access network.";
        }
        enum "non-broadcast" {
          description
          "Specify Non-Broadcast Multi-Access (NBMA) network.";
        }
        enum "point-to-multipoint" {
          description
          "Specify point-to-multipoint network.";
        }
        enum "point-to-point" {
          description
          "Specify point-to-point network.";
        }
      }
      description
        "Link access type.";
    }

    typedef multicast-source-ipv4-addr-type {
      type union {
        type enumeration {
```

```
          enum '*' {
            description
            "Any source address.";
          }
        }
        type inet:ipv4-address;
      }
      description
        "Multicast source IP address type.";
    }

    typedef multicast-source-ipv6-addr-type {
      type union {
        type enumeration {
          enum '*' {
            description
            "Any source address.";
          }
        }
        type inet:ipv6-address;
      }
      description
        "Multicast source IP address type.";
    }

    typedef timer-multiplier {
      type uint8;
      description
        "The number of timer value intervals that should be
         interpreted as a failure.";
    }

    typedef timer-value-seconds16 {
      type union {
        type uint16 {
          range "1..65535";
          }
        type enumeration {
          enum "infinity" {
            description "The timer is set to infinity.";
          }
          enum "no-expiry" {
            description "The timer is not set.";
          }
        }
      }
      units seconds;
      description "Timer value type, in seconds (16 bit range).";
```

```
     }

     typedef timer-value-seconds32 {
       type union {
         type uint32 {
           range "1..4294967295";
           }
         type enumeration {
           enum "infinity" {
             description "The timer is set to infinity.";
           }
           enum "no-expiry" {
             description "The timer is not set.";
           }
         }
       }
       units seconds;
       description "Timer value type, in seconds (32 bit range).";
     }

     typedef timer-value-milliseconds {
       type union {
         type uint32{
           range "1..4294967295";
           }
         type enumeration {
           enum "infinity" {
             description "The timer is set to infinity.";
           }
           enum "no-expiry" {
             description "The timer is not set.";
           }
         }
       }
       units milliseconds;
       description "Timer value type, in milliseconds.";
     }
   }
   <CODE ENDS>
```

4. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

```
----------------------------------------------------------------------
URI: urn:ietf:params:xml:ns:yang:ietf-routing-types
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
----------------------------------------------------------------------
```

This document registers the following YANG modules in the YANG Module Names registry [RFC6020]:

```
----------------------------------------------------------------------
name:          ietf-routing-types
namespace:     urn:ietf:params:xml:ns:yang:ietf-routing-types
prefix:        rt-types
reference:     RFC XXXX
----------------------------------------------------------------------
```

5. Security Considerations

This document defines common data types using the YANG data modeling language. The definitions themselves have no security impact on the Internet, but the usage of these definitions in concrete YANG modules might have. The security considerations spelled out in the YANG specification [RFC7950] apply for this document as well.

6. Acknowledgements

The Routing Area Yang Architecture design team members included Acee Lindem, Anees Shaikh, Christian Hopps, Dean Bogdanovic, Ebben Aries, Lou Berger, Qin Wu, Rob Shakir, Xufeng Liu, and Yingzhen Qu.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <http://www.rfc-editor.org/info/rfc7950>.

7.2. Informative References

[IEEE754] IEEE, "IEEE Standard for Floating-Point Arithmetic", IEEE Std 754-2008, August 2008.

[I-D.ietf-bfd-yang] Zheng, L., Rahman, R., Networks, J., Jethanandani, M., and G. Mirsky, "Yang Data Model for Bidirectional Forwarding Detection (BFD)", draft-ietf-bfd-yang-03 (work in progress), July 2016.

[I-D.ietf-idr-bgp-model] Shaikh, A., Shakir, R., Patel, K., Hares, S., D'Souza, K., Bansal, D., Clemm, A., Zhdankin, A., Jethanandani, M., and X. Liu, "BGP Model for Service Provider Networks", draft-ietf-idr-bgp-model-02 (work in progress), July 2016.

[I-D.ietf-ospf-yang] Yeung, D., Qu, Y., Zhang, Z., Bogdanovic, D., and K. Koushik, "Yang Data Model for OSPF Protocol", draft-ietf-ospf-yang-05 (work in progress), July 2016.

[I-D.ietf-pim-yang] Liu, X., McAllister, P., Peter, A., Sivakumar, M., Liu, Y., and f. hu, "A YANG data model for Protocol-Independent Multicast (PIM)", draft-ietf-pim-yang-03 (work in progress), October 2016.

[I-D.ietf-teas-yang-rsvp] Beeram, V., Saad, T., Gandhi, R., Liu, X., Shah, H., Chen, X., Jones, R., and B. Wen, "A YANG Data Model for Resource Reservation Protocol (RSVP)", draft-ietf-teas-yang-rsvp-04 (work in progress), October 2016.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <http://www.rfc-editor.org/info/rfc3209>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <http://www.rfc-editor.org/info/rfc4364>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <http://www.rfc-editor.org/info/rfc5880>.

7.3. URIs

[1] http://www.iana.org/assignments/address-family-numbers/address-family-numbers.xhtml

Authors' Addresses

Xufeng Liu
Kuatro Technologies
8281 Greensboro Drive, Suite 200
McLean VA 22102
USA

EMail: xliu@kuatrotech.com

Yingzhen Qu
Cisco Systems
170 West Tasman Drive
San Jose CA 95134
USA

EMail: yiqu@cisco.com

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513
USA

EMail: acee@cisco.com

Christian Hopps
Deutsche Telekom

EMail: chopps@chopps.org

Lou Berger
LabN Consulting, L.L.C.

EMail: lberger@labn.net